

# Universidad de Alcalá

## Escuela Politécnica Superior

Grado de Ingeniería Informática

### Trabajo Fin de Grado

Sistema de recreación visual mediante reconocimiento de voz

**Autor:** Beatriz Gámez San Andrés

**Tutor/es:** Ana Castillo Martínez

2020



UNIVERSIDAD DE ALCALÁ  
Escuela Politécnica Superior

**Grado en Ingeniería Informática**

Trabajo Fin de Grado  
Sistema de recreación visual mediante reconocimiento de  
VOZ

**Autor:** Beatriz Gámez San Andrés

**Tutor/es:** Ana Castillo Martínez

**TRIBUNAL:**

**Presidente:**

**Vocal 1º:**

**Vocal 2º:**

**FECHA:** 19/02/2020



## Agradecimientos

---

Este proyecto no habría sido posible sin el apoyo incondicional de mi familia, pareja y amigos.

Gracias a todos por estar conmigo independientemente de las circunstancias y de mi estado de ánimo, así como por ayudar a cambiarlo cuando ha sido necesario, por escucharme y recordarme en determinados momentos no olvidar los caminos que he decidido tomar.

He de agradecer por todo el tiempo dedicado tanto a la búsqueda de información, a superar las distintas dificultades que se han presentado, como a comprender la idea que yo tenía en la cabeza y a plasmarla en un proyecto de fin de grado a mi tutora Ana Castillo Martínez. También agradecerla en todo momento la comprensión de las circunstancias y la ayuda prestada frente a ellas.

Agradecer al colegio Calasanz por permitirme realizar el estudio de funcionalidad del sistema en un ámbito real y en especial a Gema Gómez Huguet y Pepa Burillo Chillón por compartir conmigo parte de su trabajo. A Pepa Burillo Chillón por permitirme utilizar sus ilustraciones para crear un mejor estímulo visual para los niños y a Gema Gómez Huguet por el asesoramiento realizado en cuanto a educación infantil, los hábitos de lectura y aprendizaje de los mismos; y por permitirme utilizar sus textos convirtiendo este trabajo de fin de grado en un sistema realmente aplicable y con los medios necesarios para cumplir con el objetivo establecido.



Resumen .....	10
Abstract .....	11
1. Introducción.....	12
2. Objetivo .....	14
3. Estado del arte .....	15
3.1. Juegos de comprensión lectora mediante cumplimentación de huecos .....	15
3.2. Cuentos interactivos .....	15
3.3. Audiolibros.....	16
3.4. Libros físicos .....	17
3.5. Asistentes virtuales y herramientas similares a la desarrollada.....	17
4. Desarrollo .....	19
4.1 Herramientas empleadas .....	19
4.1.1. Herramientas para el asistente virtual .....	19
4.1.2. Herramientas para el desarrollo de la aplicación Android .....	25
4.2 Arquitectura de la aplicación.....	25
4.2.1. Almacenamiento de datos .....	26
4.2.2. Estructura del asistente virtual .....	27
4.3 Implementación.....	29
4.3.1. Implementación del asistente virtual.....	29
4.3.2. Implementación de la aplicación y base de datos .....	31
4.4 Planificación del proyecto .....	34
5. Estudio de la herramienta en colegios .....	37
6. Conclusiones .....	40
7. Trabajo futuro.....	41
8. Coste del proyecto .....	42
9. Bibliografía.....	43

## Índice de Imágenes

Fig. 1. Documentación gráfica de juegos de relleno de huecos .....	15
Fig. 2. Documentación gráfica de cuentos interactivos.....	16
Fig. 3. Representación gráfica de los audiolibros .....	16
Fig. 4. Documentación gráfica de cuentos con estímulos .....	17
Fig. 5. Página de acceso a la herramienta Dialogflow .....	20
Fig. 6. Creación de intents en Dialogflow.....	20
Fig. 7. Diseño de Intent en Dialogflow .....	21
Fig. 8. Integraciones de plataformas con Dialogflow .....	22
Fig. 9. Documentación gráfica de un intento de integración.....	23
Fig. 10. Herramienta training de Dialogflow .....	23
Fig. 11. Herramienta History de Dialogflow.....	24
Fig. 12. Herramienta Analytics de Dialogflow .....	24
Fig. 13. Flujo de datos de la aplicación.....	26
Fig. 14. Interfaz MainActivity.....	32
Fig. 15. Interfaz Lectura.....	33
Fig. 16. Interfaz Lectura evolución .....	33
Fig. 17. Representación datos niños de cinco años .....	38
Fig. 18. Representación datos niños de seis años.....	38
Fig. 19. Representación datos de niños de siete años.....	38



## Índice de tablas

---

Tabla 1. Representación de costes.....	42
--	----

## **Resumen**

El trabajo que se expone a continuación relata el ámbito en el que se aplicará el sistema llamado ReadBook, los tipos de sistemas que existen hoy en día en el mercado, el diseño del sistema, el desarrollo de este y cómo se puede emplear, así como mejorar, en un futuro.

El sistema es una aplicación para Android que se compone de un asistente virtual, una base de datos y una interfaz java.

El objetivo con el cual se plantea este proyecto es la creación de un sistema funcional que ayude a los padres a crear un hábito de lectura desde una edad temprana a los niños. Con este fin se ha consultado con expertos en educación infantil y el proyecto ha tomado las siguientes características: una interfaz de usuario simple e intuitiva, un manejo de la aplicación de manera oral siempre que ha sido posible, simplicidad en los textos y estímulos visuales mediante imágenes.

El sistema permite leer cuentos, cada uno de ellos contiene múltiples imágenes, todas ellas asociadas a una frase. La frase será repetida hasta que se lea correctamente con el fin de mejorar la lectura del niño y se obtendrá un refuerzo positivo al leer correctamente y un avance en la historia que se está leyendo.

## **Palabras Clave**

Asistente virtual, SQLite, Android Studio, educación y hábito de lectura.

## **Abstract**

The work presented below describes the area in which the system called ReadBook will be applied, the types of systems that exist today in the market, the design of the system, the development of this system and how it can be managed, as well as how it can be improved in the future.

The system is an Android application that consists of a virtual assistant, a database and a java interface.

This Project has been realized with the objective of developing a functional system that helps parents to create a habit of reading from an early age to children. In order to create a good application, experts in early childhood education have been consulted and the project has taken the following characteristics: a simple and intuitive user interface, an orally application management whenever it's possible, simplicity in the texts and visual stimulus by images.

The system allows you to read stories, each one contains multiple images, all of them associated with a phrase. The phrase will be repeated until it is correctly read in order to improve the child's reading. There will be a positive reinforcement when the text is read correctly and an advance in the story will be perform.

## **Key words**

Virtual assistant, SQLite, Android Studio, education and Reading habits.

# 1. Introducción

Todos hemos visto en la calle a esos padres que dejan a los niños el teléfono, ya sea en un restaurante, cuando van a casa de familiares o cuando no se desea que molesten. Es un hecho que vivimos en la era de la tecnología y cada día más, los niños están rodeados de múltiples estímulos que no les permiten centrar su atención en algo durante mucho tiempo. Estos estímulos son principalmente el movimiento, el sonido y color, por lo que si se quiere captar la atención de los más pequeños, la forma más efectiva es utilizando estos estímulos.

Debido a la gran pérdida del hábito de lectura en general, se han visto realizados múltiples estudios para buscar la causa de esta pérdida. La mayoría de los estudios realizados concluyen en diferentes causas, pero casi todos ellos coinciden en la presencia de teléfonos, tablets y tecnologías que capturan la atención y la mantienen más de lo que haría un libro. Al captar más fácilmente la atención de las personas, hace que estas inviertan más tiempo en los aparatos tecnológicos que en cualquier otra actividad.

Este proyecto consiste en la creación de un sistema para fomentar o crear los hábitos de estudio en los niños que están comenzando a leer o que están terminando de aprender a hacerlo. Con la propuesta que se plantea en este proyecto se busca unificar de alguna forma, la tecnología y esos estímulos que nos hacen perder actividades beneficiosas para el intelecto, con dichas actividades, llegando a crear actividades que no sólo sean más amenas, sino que nos llamen la atención y produzcan esa necesidad de uso propia de los aparatos tecnológicos.

El proyecto está orientado a niños que ya han comenzado a leer frases, pero contiene otros cuentos más avanzados para ayudar a la creación o mejora de un hábito de lectura. Es por ello que para su realización se ha pedido asesoramiento a profesores de educación infantil, con la formación y conocimientos necesarios para guiar a los niños al aprendizaje de la lectura. El hecho de utilizar la actitud positiva hacia estas aplicaciones en general del público al que va dirigido, los niños, le proporcionará al proyecto a desarrollar la capacidad de llegar fácilmente a sus destinatarios.

Las tecnologías que se han decidido usar en el proyecto son tecnologías que ya están incluidas en el día a día de los niños, y por lo tanto, no causarán una incomodidad frente a algo desconocido por parte de los padres o de los niños. Estos motivos son los que han llevado a usar un asistente virtual con tecnología de Google en la nube al que se accede desde una aplicación, dejando como única entrada y salida un teléfono o Tablet. Estos dispositivos están asociados a actividades lúdicas como pueden ser los juegos o la visualización de videos, a los que los niños ya están acostumbrados y por lo tanto les resultan atractivos, ya que los tienen asociados a la diversión y el entretenimiento. Se va a aprovechar este hecho con el fin de realizar una aplicación que aproveche los beneficios que nos proporcionan estos dispositivos y a su vez permita la enseñanza de una parte muy importante del aprendizaje infantil como es la lectura.

El proyecto consiste en un asistente virtual que se encargará de convertir las frases leídas por los niños en información tratable desde la aplicación a la que se conectará, una base de datos SQLite donde se almacenaran los cuentos que va a tener la aplicación y el contenido

textual y visual necesario para la creación de la aplicación. El desarrollo de la aplicación se ha llevado a cabo en Android Studio utilizando la versión 9.0 o Pie y como lenguaje de programación se ha utilizado java.

Para validar su utilidad de cara mejorar hábitos de lectura en los niños, se ha realizado un estudio de la aplicación desarrollada por medio de su implantación en el colegio Calasanz de Alcalá de Henares. Compartiendo los cuentos que se emplean en dicha institución escolar con el proyecto y probando la usabilidad de la aplicación en función de las distintas edades, se han podido analizar los datos obtenidos para sacar conclusiones de forma práctica beneficiosas para el análisis de las posibles mejoras a futuro para llevar la aplicación al mercado.

El documento se encuentra estructurado de la siguiente manera: primero se establecerá una descripción del objetivo del proyecto desarrollado. Posteriormente se procederá a la enumeración de las distintas plataformas existentes para los mismos propósitos que se encuentran en el mercado, de esta forma se podrán comparar los distintos beneficios proporcionados por cada uno de los existentes con respecto al proyecto que se está presentando en este documento. El desarrollo que le sigue explica detalladamente el proceso de creación de la aplicación, primero las herramientas que se han empleado en el desarrollo, luego la estructura que se ha empleado, así como la conexión entre las distintas herramientas desarrolladas en el apartado anterior, más tarde se explica en detalle la implementación del proyecto y por último la planificación que se ha llevado a cabo durante todo el desarrollo. Posteriormente se procederá a analizar las conclusiones a las que nos lleva la creación de esta aplicación y seguido, el trabajo que se podría realizar a futuro para que el proyecto mejorase. Más tarde se exponen los costes del proyecto, donde se simulan casos factibles de balances de dinero. Se adjunta el estudio realizado en el colegio, sobre el uso de la aplicación en distintas edades. Para finalizar se expone la documentación consultada para llevar a cabo todo el desarrollo del proyecto.

## **2. Objetivo**

El proyecto tiene como objetivo la utilización de los estímulos que proporciona la tecnología, que a menudo elimina actividades necesarias o beneficiosas para los niños, para la creación o afianzamiento de un hábito de lectura en los más pequeños.

La creación de un hábito de lectura no sólo proporciona una mayor fluidez en el desarrollo de las actividades escolares, sino que además fomenta un correcto uso del lenguaje. Por lo que esta aplicación tiene como objetivo complementar los estudios, facilitando el aprendizaje de los niños y el repaso de lo aprendido en la escuela.

Se pretende mezclar el refuerzo positivo en el aprendizaje con los estímulos visuales de imágenes coloridas, con tecnologías incorporadas en la vida cotidiana y con los estímulos auditivos con el fin de mantener la atención del niño en una actividad que puede suponer un reto.

Se intenta proporcionar un entorno que resulte familiar para padres e hijos, de manera que se facilite la utilización. Se ha creado una interfaz sencilla e intuitiva para facilitar el uso, ya que se considera que los usuarios pueden abarcar desde niños que acaban de comenzar a leer hasta niños capaces de leer correctamente y con fluidez. Debido a la consideración de la primera clase de usuarios la aplicación puede ser utilizada mediante voz casi en su totalidad.

Para poder conseguir el objetivo establecido se han tomado como medidas la consulta con gente dedicada a la educación infantil, impartiendo clases en las que deben enseñar a leer a los alumnos. Se ha contactado con los autores de cuentos creados especialmente para el aprendizaje de la lectura y se han tomado las fases más avanzadas de ellos para su incorporación en la aplicación, dando una utilidad real y dando la posibilidad de realizar un estudio del uso de la aplicación y si realmente se alcanzan los objetivos que se han expuesto anteriormente y en qué medida.

### 3. Estado del arte

Cuando se plantea el trabajo que se quiere realizar se han consultado los diferentes productos que se encuentran previamente desarrollados cuyo fin es ayudar de una forma u otra a comprender diferentes historias mediante la lectura. En la actualidad existen múltiples herramientas para complementar el aprendizaje o facilitarlo en mayor o menor medida, de las cuales se utiliza de forma general la comprensión lectora, completando textos o escuchando audios de la historia relatada. También se encuentran herramientas que emplean diferentes estímulos para mantener la atención del lector. A continuación, se van a explicar más detalladamente algunas de ellas.

Se procede a la explicación de algunas de las diversas herramientas que emplean aparatos tecnológicos como plataforma de despliegue.

#### 3.1. Juegos de comprensión lectora mediante cumplimentación de huecos

Los primeros sistemas que se vieron como complemento a la educación y el aprendizaje eran pequeños juegos donde se daban opciones para rellenar huecos. En este tipo de juegos se requería la asociación y comprensión del contexto con el lenguaje y se aplica en niños que ya saben leer. También es utilizado para el aprendizaje de nuevos idiomas por los libros de texto empleados en los colegios.



Fig. 1. Documentación gráfica de juegos de relleno de huecos

#### 3.2. Cuentos interactivos

Posteriormente aparecieron los cuentos interactivos donde la historia se va desarrollando ya sea de manera sólo textual o con complementación visual. El lector debe

comprender la historia que se está relatando para luego decidir cómo desea que continúe el argumento. Al llegar al final del texto seleccionado por la persona que crea la aplicación se crea la posibilidad de elegir entre diversas ramas que tomar, creando múltiples cuentos en uno sólo. Este tipo de herramienta también se puede encontrar en formato físico de libro dónde el lector debe avanzar manualmente a la página que le indique el libro en función de la opción que desee elegir.

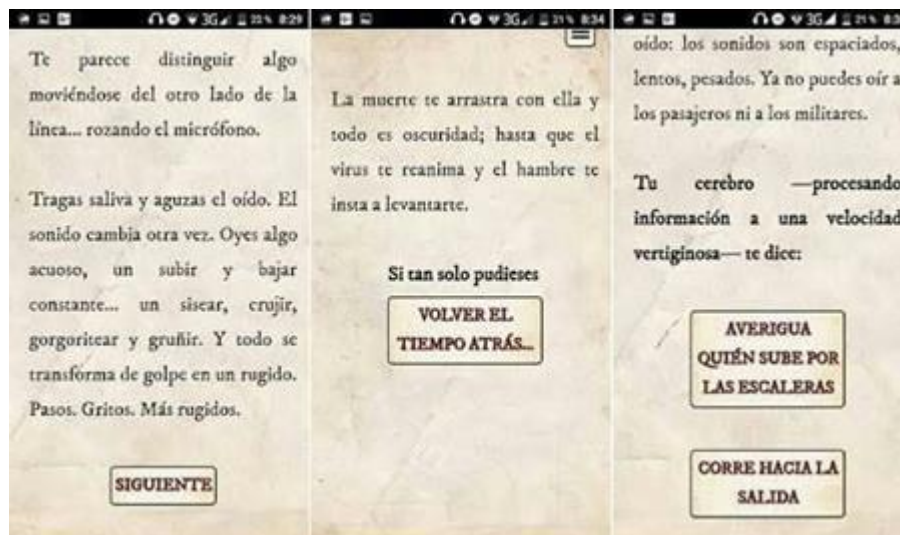


Fig. 2. Documentación gráfica de cuentos interactivos

### 3.3. Audiolibros

Otra de las herramientas utilizadas son los audiolibros, que ya emplean tecnologías más complejas como puede ser la inteligencia artificial. Estos sistemas van leyendo el libro que haya sido elegido y sólo es necesario ponerse unos cascos para disfrutar de la historia. Si bien esta última opción es llamativa y la gente la utiliza debido a la falta de tiempo en el día a día, no contribuye a la ampliación del vocabulario o a una mejor gramática, ya que es el sistema el que narra. Aunque en su mayoría se utilizan para realizar otras actividades a la vez que se escucha el libro, también se utiliza para aprender la pronunciación de otros idiomas mientras vas siguiendo la lectura a la par que la escuchas.



Fig. 3. Representación gráfica de los audiolibros



Se explica a continuación en qué se basan las herramientas en formato de libro físico, que tratan de emplear diferentes modificaciones y conocimientos del comportamiento infantil.

### 3.4. Libros físicos

Para los más pequeños se utilizan libros que tengan olores, muchos colores, que tengan despleables e incluso con diferentes texturas. Con el fin de usar los estímulos que mantienen la atención de los más pequeños en el objeto, al recibir todos los estímulos del objeto concentran su atención en este, en vez de en los posibles estímulos por separado que puedan llegarles.



Fig. 4. Documentación gráfica de cuentos con estímulos

### 3.5. Asistentes virtuales y herramientas similares a la desarrollada

Los asistentes virtuales son herramientas que se están usando principalmente para la realización de controles de diversas plataformas mediante voz. Nos permiten añadir elementos a nuestra lista de la compra, apagar luces, acceder a noticias, subir persianas... pero ¿qué hacen realmente estos asistentes?

Los asistentes graban en todo momento los sonidos que se producen a su alrededor, a la espera de una palabra que sea identificada como comienzo de una interacción. En el caso de Alexa es el propio nombre que se le ha dado al dispositivo, en el caso del asistente de Google la frase "OK, Google". Cuando se detectan estas conexiones iniciales, el sistema va analizando lingüísticamente y aplicando el contexto, para detectar diversas formas de decir algo que coincidan con los intents que tiene definidos. La mayoría de estas actividades que se pueden

realizar son triggers que desencadenan una acción del dispositivo al que esté conectado. El asistente tendrá las funciones diseñadas por las empresas y no ha sido posible, hasta hace poco, que se han facilitado las API's a los desarrolladores, cuando se han comenzado a realizar pequeñas funcionalidades distintas a las originales.

Para muchas de estas grandes empresas, sus herramientas permiten hacer cosas muy limitadas, ya que están en una fase muy temprana de creación de API. Muchas utilidades se ven mermadas, sin acceso o muy propensas a errores, debido a que se encuentran en fase beta. En el caso de la herramienta IFTTT sólo puedes realizar aquellas funcionalidades que ya se puede de fábrica, pero aplicándolas en función de tu código y con escasas plataformas compatibles. En el caso de Dialogflow, se permite un tratamiento de datos a más bajo nivel, ya que puedes recibir el json con la información de la interacción y tratarlo posteriormente en tu aplicación, si bien es la principal característica que hace a esta herramienta más útil, al estar en fase beta, no siempre detecta correctamente los acentos y la pronunciación.

Estos productos se han puesto muy de moda ya que, gracias a la inteligencia artificial, son más fácilmente incorporables a la vida diaria. Con una breve conversación o comando oral, puedes realizar diversas acciones facilitando la realización de actividades rutinarias o programándolas para distintas horas. Debido a todo lo explicado, estas herramientas se están tratando de incorporar a otros ámbitos donde produzcan una mejora.

Una de las aplicaciones que se le está dando es la traducción de idiomas, ya que se puede realizar una consulta como: “Cómo se dice linterna en alemán” y se recibirá una respuesta indicando la traducción que encuentre en internet.

En el caso de Alexa y de Google cuenta con una mayor cantidad de actividades para uso educativo, te lee libros, te calcula las cuentas que preguntes en voz alta, hace traducciones simples a otros lenguajes, como todos los asistentes cuenta con búsquedas en internet por lo que pueden proporcionar información más rápido, pero menos contrastada.

Actualmente existen aplicaciones para la gestión de centros educativos de la mano del asistente de voz de Alexa, pero como ayuda al aprendizaje se dan sólo las aplicaciones expuestas en el párrafo anterior.

## **4. Desarrollo**

En el apartado de desarrollo de este proyecto se van a exponer cómo se ha desarrollado la aplicación. Para ello se comenzará explicando las herramientas que se han usado para la implementación de la aplicación, del asistente virtual y de la base de datos. Posteriormente se explicará la estructura de la aplicación y la interconexión entre las herramientas del apartado anterior. Para finalizar se narra la implementación que se ha realizado para la aplicación y la planificación que se ha llevado durante todo el desarrollo del proyecto.

### **4.1 Herramientas empleadas**

Las herramientas que se van a emplear para realizar el desarrollo se han escogido tras analizar los requisitos que debían tener para cumplir con las necesidades funcionales del proyecto. En primer lugar, se va a explicar la elección de la herramienta para crear el asistente virtual, y posteriormente aquellas herramientas necesarias para desarrollar la aplicación móvil.

#### **4.1.1. Herramientas para el asistente virtual**

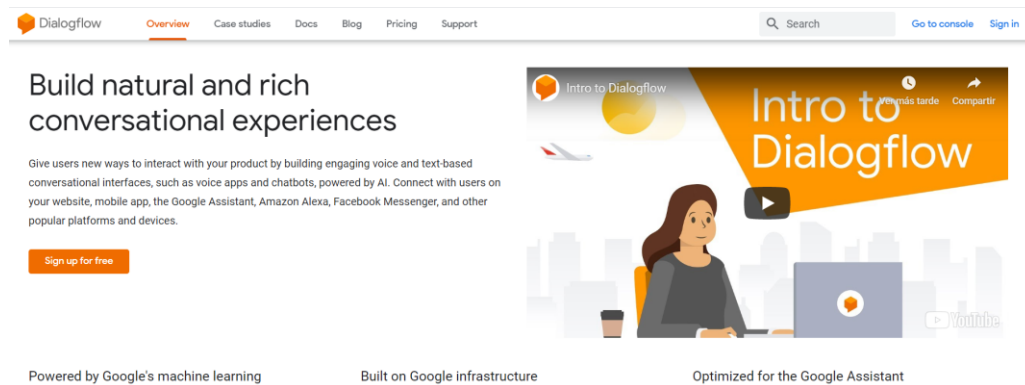
Para el desarrollo de la aplicación se necesitaba un reconocimiento de voz que permitiese el traspaso a escritura con facilidad. Para ello se barajaron diferentes posibilidades como la de utilizar el micrófono de la plataforma que se eligiese para procesar el audio. Sin embargo, debido al aprendizaje, aunque lento, de la voz de los asistentes virtuales se decidió emplear uno que se encontrase ya en desarrollo. El primer problema encontrado fue la falta de información sobre las maneras de integrar las herramientas. En algunos casos las herramientas únicamente proporcionaban servicios muy concretos como era encender y apagar diversas funcionalidades del teléfono (bluetooth, wifi, sonido...), así como abrir únicamente la aplicación Google Maps.

Debido a que los asistentes existentes no se podían emplear como tal porque ninguno de ellos contenía la API necesaria para devolver lo entendido y la compatibilidad con la aplicación Android. Se ha requerido una búsqueda de herramientas a nivel más técnico de lo que ofrecían los asistentes por sí solos, por eso se llega a la herramienta en fase de pruebas que permite el uso del asistente de Google y permite el tratamiento de los datos mediante una API de conexión entre la plataforma y el asistente.

Tras analizar diferentes opciones, se decidió utilizar la herramienta Dialogflow, la cual es una herramienta de Cloud proporcionada por Google. Entre los motivos para seleccionar esta herramienta se encuentran la cantidad de manuales, la información para desarrolladores y la funcionalidad de devolver una respuesta en formato JSON legible por Android como Hashmap. Cabe destacar que esta herramienta se encuentra en gran parte en fase beta y se trata de la primera versión de la misma, lo que hace el tratamiento del desarrollo más arduo y la documentación respecto al mismo es más escasa.

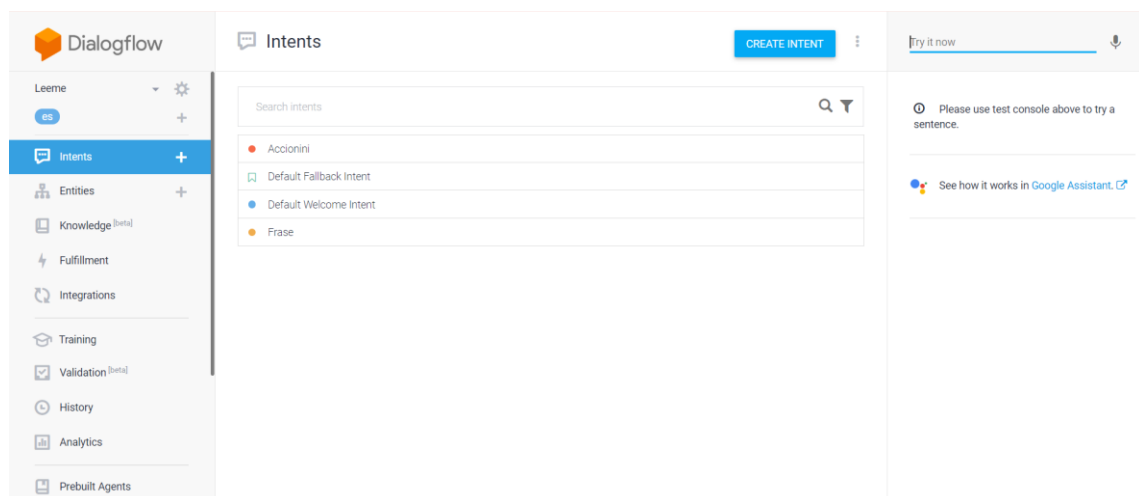
Esta herramienta proporciona la funcionalidad de crear un asistente virtual, modificando las respuestas que se quieren obtener en función de las frases de entrada y extrayendo de las

frases los fragmentos que son relevantes para su análisis. Se encuentra en la nube por lo que se pueden conectar múltiples dispositivos debido a la escalabilidad que proporciona. Es compatible con Android por lo que para el desarrollo del proyecto era la mejor de las elecciones, además de ser ampliamente intuitiva y gratuita.



**Fig. 5. Página de acceso a la herramienta Dialogflow**

Al abrir esta herramienta nos encontramos con la siguiente ventana que nos permitirá navegar por las diversas funcionalidades:



**Fig. 6. Creación de intents en Dialogflow**

Entre las opciones se encuentra la opción “Intents”, los cuales son agrupaciones de frases o palabras que nos van a proporcionar el mismo resultado. En este caso, para el proyecto que se quiere llevar a cabo, la aplicación cuenta con un Intent que hemos creado al que hemos llamado Accionini. Accionini es una estructura de tipo Intent que se asociará con las acciones iniciales de la pantalla de la aplicación que se va a crear, donde contemplaremos las posibilidades de que se diga sólo la acción (ej: *leer*) o la acción dentro de una frase (ej: *quiero leer*). Para entrenar al asistente debemos contemplar el mayor número de posibilidades que consideremos que se pueden usar para llamar a esa acción. Se introducen varias frases y se asocia a las palabras leer, continuar, cargar un tipo concreto que hemos creado.

Una vez que la acción es detectada como tipo accion, se produce la respuesta que se haya designado o se elige entre alguna de ellas. En este caso se obtendrá accediendo a \$accion o procediendo a \$accion.

The image shows the Dialogflow Intent Designer interface. At the top, there's a section for 'Intent name'. Below it are tabs for 'Contexts', 'Events', and 'Training phrases'. The 'Training phrases' tab is active, showing a text input field with a microphone icon and the placeholder 'Add user expression'. Below this is the 'Action and parameters' section, which includes a field for 'Enter action name' and a table for parameters. The table has columns for 'REQUIRED', 'PARAMETER NAME', 'ENTITY', and 'VALUE'. There's a '+ New parameter' link below the table. The 'Responses' section is below, showing a 'DEFAULT' response with a text input field for 'Enter a text or SSML response' and an 'ADD RESPONSES' button. At the bottom, there's a checkbox for 'Set this intent as end of conversation' and a 'Fulfillment' section.

**Fig. 7. Diseño de Intent en Dialogflow**

Cuando se crea un nuevo Intent, la herramienta permite realizar configuraciones respecto a él, permitiendo darle un nombre que posteriormente se utilizará para evaluación de la respuesta, permitiendo asociar un evento a la identificación de las frases, se piden frases para entrenar el agente con las que se consigue si sólo se quiere analizar una palabra de la frase modificar dicha palabra en función de los diversos contextos que tenga. Cuantas más frases se introduzcan más fácil será para el asistente comprender los contextos y tonos del lector.

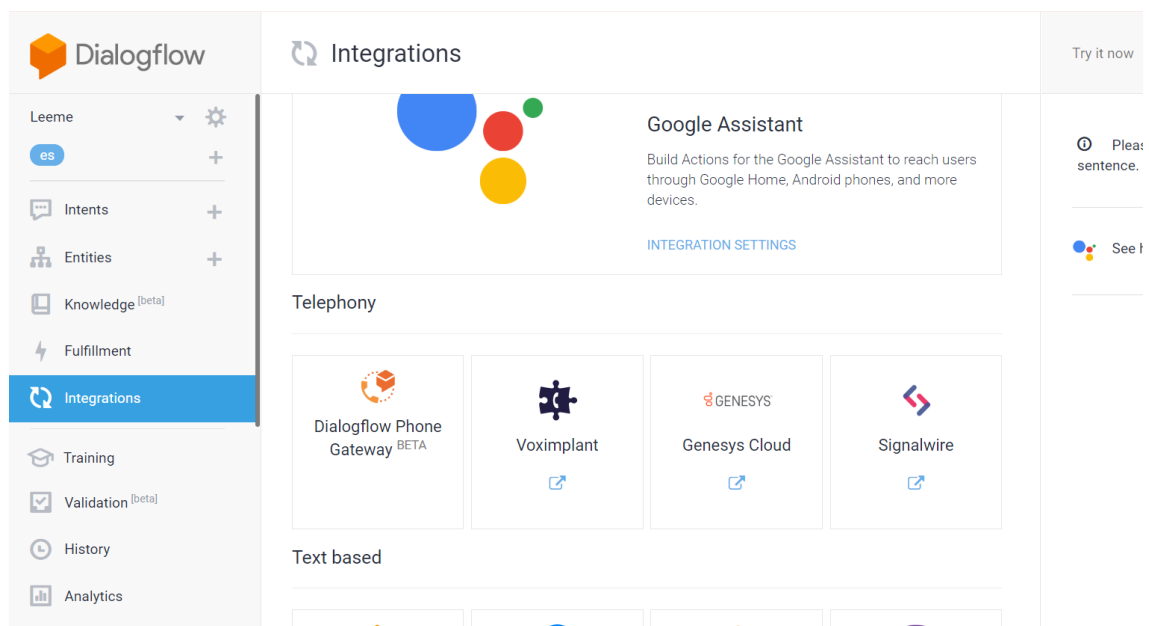
Debido a la explicación anterior, podemos razonar que la compresión de frases sueltas sin ningún tipo de conexión lingüística que se van a utilizar para el desarrollo de este proyecto no son la mejor forma de entrenar al agente. Dado que te devuelve dentro del JSON sólo la parte de la frase que se tenga declarada como específica de un tipo, se tomó como decisión de desarrollo incluir un Intent frase que contenga como frases de entrenamiento las frases de los cuentos. Esto da lugar a pequeños errores por entonaciones en los que el asistente no identifica bien entre palabras muy similares en cuanto a pronunciación, pero muy diferentes en cuanto a significado y con leves diferencias de escritura, como es el caso de “*tardó mucho tiempo*” que lo identifica como “*tardo mucho tiempo*” dando lugar a confusión en la aplicación.

Dado que la aplicación está orientada a niños que apenas han empezado a leer y los cuentos no deben tener excesiva complejidad, se ha tomado la decisión de modificar las frases dentro de lo posible para evitar palabras que suenen igual, eliminando la tonalidad de la palabra como referencia, y se escriban diferente por una tilde.

En el apartado de acciones y parámetros se le da nombre a la acción que se va a realizar, con los parámetros definimos aquellas cosas que se deben decir para que se lleve a cabo una acción. Los parámetros se pueden hacer optativos, si no están y se identifica parte de la frase de entrenamiento se ejecuta la acción, u obligatorios, aquellos parámetros que se definan como obligatorios harán que la acción se lleve a cabo sólo y exclusivamente en caso de que aparezcan todos aquellos que hayan sido definidos como obligatorios dentro de la frase.

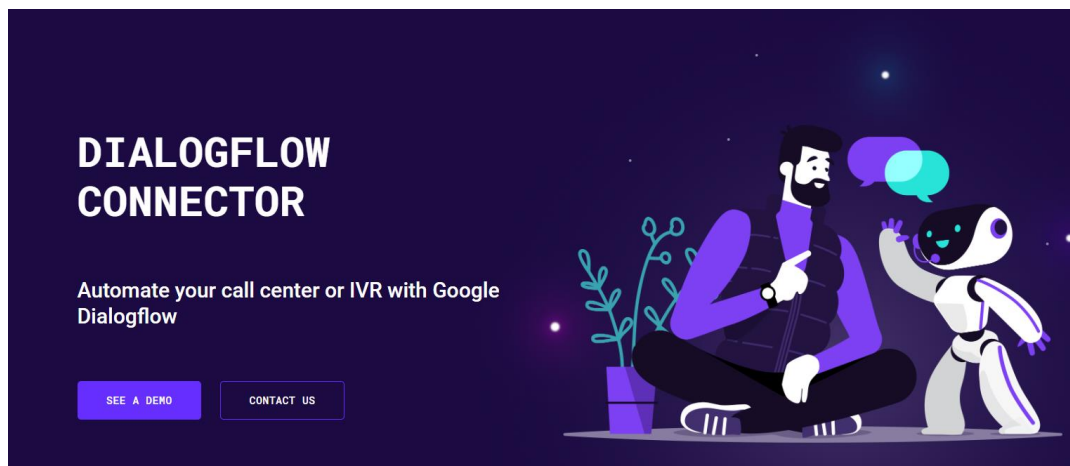
El siguiente apartado de la aplicación, titulada Responses, nos permite crear las respuestas que vamos a querer que nos devuelva el asistente cuando identifique alguna de las frases definidas en este Intent. Se puede meter una única respuesta en caso de que queramos que sea siempre la misma, o incluir varias de ellas con el fin de que el asistente vaya variando entre ellas y dé mayor sensación de conversación. En la parte inferior de este apartado se nos permite decidir si queremos que la conversación finalice al identificar nuestro Intent y devolver una respuesta. En el caso de la aplicación que se está creando en este proyecto no nos interesa crear ningún Intent con estas características ya que lo que queremos es que en caso de fallo se pueda releer de nuevo.

Fulfillment, herramienta proporcionada por Dialogflow, nos permite realizar acciones de POST sobre cualquier software que creemos al que se le pueda incorporar la API de Dialogflow. Esto nos permite conectarlo. También se puede modificar a nivel más bajo como se quiere que se realice este intercambio de información.



**Fig. 8. Integraciones de plataformas con Dialogflow**

La opción Integrations permite integrar el asistente creado en las múltiples plataformas que se ofrecen. A pesar de proporcionar gran cantidad de plataformas se trata de un servicio que aún no está disponible como tal, al pinchar en ellas permiten solicitar contactar con ellos o enviar una demo del asistente que has creado con sus funcionalidades para evaluar si lo incorporan o no, por esta razón se ha usado simplemente la configuración expuesta anteriormente, en el proyecto.



**Fig. 9. Documentación gráfica de un intento de integración**

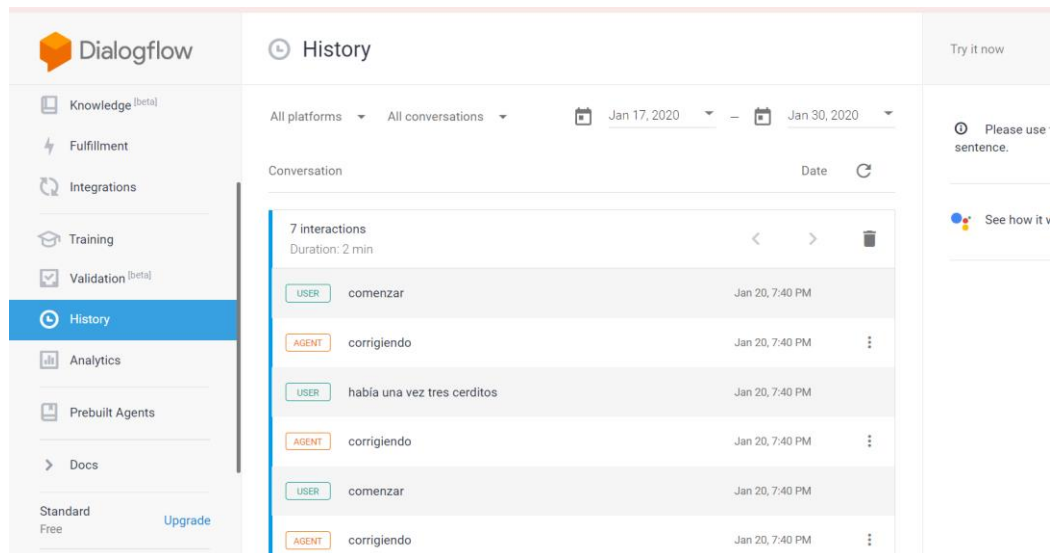
La herramienta Training permite analizar las respuestas que ha obtenido en una conversación. Con esta herramienta podemos analizar las conversaciones, qué es lo que ha entendido el asistente, qué parámetro lo está analizando, el Intent al que lo asocia. En caso de que no tenga esa información proporciona la opción de añadirlo al Intent que considera que se acerca más a lo deseado, como una frase de entrenamiento más. También permite incorporarla al default y eliminarla, lo que hace que el asistente aprenda de las conversaciones, pero debe ser guiado por un humano para ello.

Conversation	Requests	No match	Date
tiene que ser	2	0	Jan 24
ver	17	0	Jan 20
leer	14	0	Jan 20
comenzar	7	0	Jan 20
comenzar	10	0	Jan 20

**Fig. 10. Herramienta training de Dialogflow**

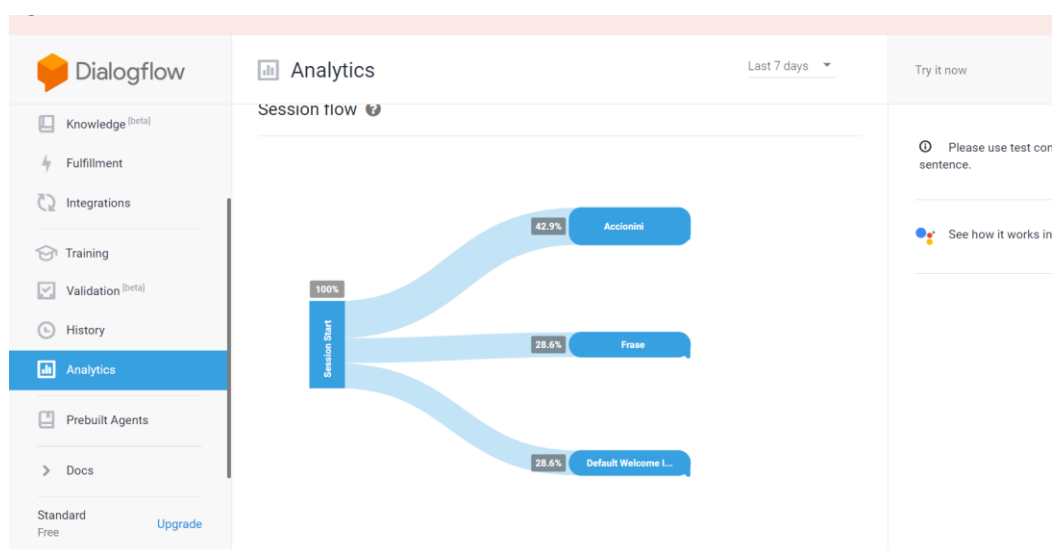
Con la herramienta History podemos acceder al historial de conversaciones que se han llevado a cabo y analizarlas a un nivel más bajo. Cuando existe una conversación el asistente guarda un registro de las interacciones, la duración del tiempo y lo que a nivel de programación es mucho más interesante, los objetos JSON enviados. Con esta herramienta no sólo puedes hacer un análisis de los fallos, de las correcciones que se deben tomar de cara a mantener la conversación que se quiere con el usuario, sino que además puedes estudiar los parámetros que te son devueltos, todos los campos que se guardan y usarlos correctamente a la hora de programar. Cabe destacar que al interconectar Dialogflow con Android, estos parámetros dejan

de tratarse como JSON y pasan a tratarse como Hashmap pero contienen los mismos campos por lo que facilita igualmente el desarrollo y tratamiento de los datos.



**Fig. 11. Herramienta History de Dialogflow**

La herramienta Analytics proporciona múltiples ejemplos visuales de los datos que va recopilando por medio de los usos de la aplicación. En este caso se pueden ver por dónde han circulado los flujos de datos, pudiendo entender que la aplicación ha sido controlada por voz aproximadamente la mitad de las veces desde el inicio de la aplicación que es donde podemos encontrar los Intent Accionini. También podemos observar que no siempre que se ha abierto la aplicación se ha llegado a leer, por lo que podemos detectar que es probable que hayamos tenido un bug o se hayan realizado pruebas sólo en la pantalla inicial, que es el caso. También observamos que se ha saludado cordialmente al asistente antes de comenzar a usar la aplicación en algunos casos. Este tipo de informaciones visuales de las que se puede sacar mucha información nos proporciona herramientas muy potentes para realizar estudios.



**Fig. 12. Herramienta Analytics de Dialogflow**



Dentro del propio Dialogflow tenemos la herramienta Docs, que contiene todos los documentos con las informaciones necesarias para llevar a cabo desarrollos, esto facilita mucho la creación de asistentes en esta herramienta y la integración con otras herramientas y/o plataformas.

#### **4.1.2. Herramientas para el desarrollo de la aplicación Android**

En cuanto a la herramienta empleada para el desarrollo de la aplicación se han tenido en cuenta múltiples posibilidades, pero se ha decidido utilizar Android Studio, debido a la compatibilidad que ofrecía con los diversos asistentes virtuales, frente al resto de herramientas posibles. Además esta herramienta proporciona múltiples facilidades para la creación de aplicaciones en plataformas, con las que los niños están ampliamente familiarizados como plataformas de teléfono y tablets. A esto se le añade, la posibilidad de llevarse la aplicación con uno mismo y tratar de aplicar el cambio de videos por lectura. Como consecuencia de todas estas características se da lugar a una aplicación más usable y altamente incorporable al mercado actual.

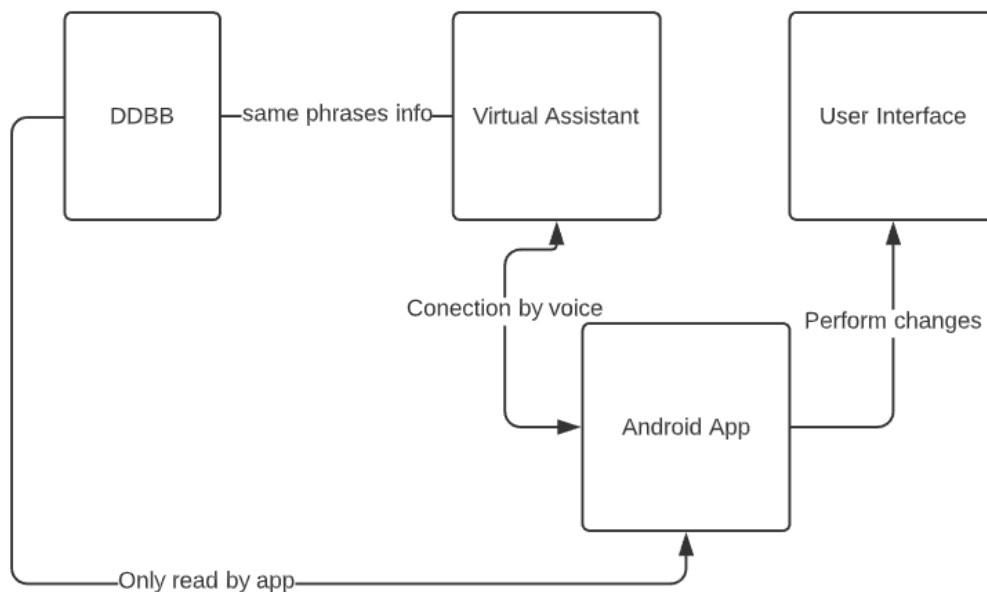
Una de las cualidades que ha hecho de Android Studio la herramienta elegida ha sido la compatibilidad con múltiples herramientas de creación de asistentes virtuales. La familiaridad con el lenguaje tan similar a Java facilita la creación debido a que es un lenguaje conocido y en el que se tiene más experiencia.

En cuanto al almacenamiento de datos, se ha optado por usar una base de datos SQLite ya que en un principio se quería guardar las imágenes de todos los cuentos dentro de la misma y como complemento se ha usado el archivo que nos permite usar Android para guardar las preferencias de usuario. Mediante las funcionalidades que esto nos proporciona, se ha conseguido guardar los progresos, continuarlos y modificarlos.

La primera opción para el almacenamiento de las imágenes era mantenerlas todas dentro de la base de datos e ir extrayéndolas en función de la necesidad de la aplicación. Se encontró un problema con los archivos gráficos, los cuales deben estar contenidos dentro de la carpeta drawable para poder ser mostrados y tratados correctamente. Como decisión de diseño para solventar el problema se incorporaron las imágenes de los cuentos a la carpeta drawable y en la base de datos se guardaron los nombres de las imágenes con las que se podría acceder a su id y así modificar la interfaz.

## **4.2 Arquitectura de la aplicación**

En la siguiente imagen se muestra el flujo de datos que se ha desarrollado en la aplicación y la interrelación entre las herramientas, backend y frontend.



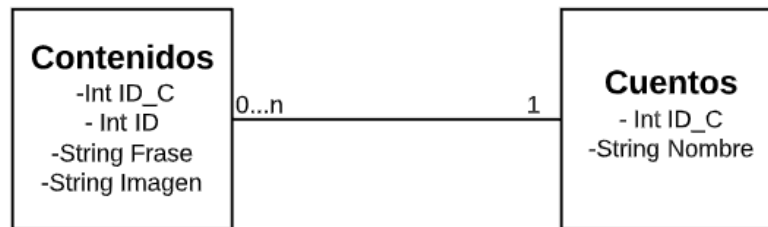
**Fig. 13. Flujo de datos de la aplicación**

La aplicación es la base del sistema, ya que es dónde realizaremos la interacción con el usuario y dónde conectaremos los distintos servicios que la complementarán. La aplicación es la que establece el inicio de la conexión con el asistente virtual, mediante la activación de un listener, quedará a la espera de la respuesta del asistente virtual que llegará en formato json por la conexión establecida. Una vez obtenida la respuesta la aplicación consultará con la base de datos si lo obtenido coincide con lo esperado, de ser así se realizará la acción deseada, ya sea avanzar a la lectura o modificar la interfaz de usuario con los nuevos datos a detectar. Una vez realizada la modificación y dada la respuesta del asistente, la aplicación retomará la conexión con el asistente en caso de ser necesario.

#### **4.2.1. Almacenamiento de datos**

Para el almacenamiento de datos se ha empleado una base de datos SQLite, la principal razón por la que se tomó esta decisión fue la capacidad de integración con Android y la necesidad de almacenar imágenes en dicha base de datos. Posteriormente y debido a que la interfaz no podía ser modificada si las imágenes no estaban contenidas dentro de la carpeta drawable, se tuvo que modificar el campo imagen de tipo Image por una de tipo Text.

A continuación, se muestra la estructura de las tablas que componen la base de datos y la información contenida en ellas para una mejor comprensión del funcionamiento.



**Fig. 14. Diagrama de entidad relación**

Como se aprecia en la imagen la estructura de la base de datos no es altamente compleja, esto se debe a que los datos que se necesitan almacenar son muy específicos. Se requiere saber qué cuento se está leyendo por lo que se crea la necesidad de la existencia de la tabla Cuentos, donde se debe almacenar el nombre. Cada uno de los cuentos, contiene frases y estas frases deben ser asociadas al cuento, por lo que surge la necesidad de crear el campo ID\_C que nos permitirá hacer dicha asociación. Las frases requieren saber la posición o página donde se encuentran por lo que se les asocia un ID dentro del cuento, además debe contener la frase que se quiere mostrar y comparar además de la imagen que se va a emplear para el estímulo visual de dicha frase. Como resultado obtenemos una base de datos simple pero a la vez muy clara y fácilmente tratable.

#### **4.2.2. Estructura del asistente virtual**

En cuanto a la estructuración del asistente virtual se han creado los siguientes Intents: Accioini y frase. Cabe destacar que se le ha dado mayor prioridad a Accionini lo que quiere decir que en caso de duda, respecto a la pertenencia de una frase a cualquiera de los dos Intents será automáticamente valorada como Accioini. Esta decisión se tomó a raíz de la necesidad de obligar a elegir una de las acciones de MainActivity sin que la corrección fallase por ser detectada como una frase perteneciente a alguno de los cuentos. En el Intent frase lo que se almacenan son todas las frases de los cuentos, lo que nos crea limitaciones a la hora de introducir nuevos cuentos, ya que no se pueden introducir de forma automatizada, sino que se deben introducir manualmente. Se cree que esto se debe a la fase en la que se encuentra la herramienta ya que contiene múltiples funcionalidades en fase beta y algunas sin poder ser utilizadas, a futuro se podrá integrar mejor y esta delimitación se espera que desaparezca.

Las características de los Intents hacen que se diferencien en las respuestas que se devuelven cuando se reconocen como en el caso de las frases de los cuentos recibiremos una respuesta diciendo bien en caso de ser detectada y en el caso de reconocer una Accionini nos devolverá una de las siguientes variantes: procediendo a \$accionini o accediendo a \$accionini. Otra de las diferencias entre los dos Intents es que Accionini se crea pensando en un lenguaje coloquial mientras que las frases se quieren detectar sólo cuando sean idénticas, por esta razón la primera de estas estructuras tiene frases de ejemplo que lo entrenan para comprender que la frase “*quiero leer un libro*” se refiere a la acción leer.

El asistente proporcionará a la aplicación mediante la API una respuesta en formato json, esta respuesta debe ser tratada como un HashMap por la aplicación y tiene la estructura que se muestra a continuación:

```
{
  "queryText": "había una vez tres cerditos",
  "action": "leyendo",
  "parameters":
  {
    "frase": "había una vez tres cerditos"
  },
  "fulfillmentText": "corrigiendo",
  "fulfillmentMessages":
  [
    {
      "text":
      {
        "text":
        [
          "corrigiendo"
        ]
      },
      "lang": "es"
    }
  ],
  "intent":
  {
    "id": "40a6b7cc-ac13-448e-a3e0-ed4d42d9f476",
    "displayName": "Frase",
    "priority": 750000,
    "events":
    [
      "Leyendo"
    ],
    "action": "leyendo",
    "parameters":
    [
      {
        "id": "ddc1b220-9f09-4042-9347-182b0061f75f",
        "displayName": "frase",
        "value": "$frase",
        "entityTypeDisplayName": "@sys.any",
        "reprompt": { }
      }
    ],
  },
}
```

```

      "messages":
      [
        {
          "text":
          {
            "text":
            [
              "bien"
            ]
          },
          "lang": "es"
        }
      ]
    },
    "intentDetectionConfidence": 1,
    "languageCode": "es",
    "slotfillingMetadata":
    {
      "allRequiredParamsPresent": true
    },
    "id": "db4ad858-bd24-45cf-ac72-4a175dd1f1b1-426bc00a",
    "sessionId": "95fa500c-75ef-4405-865c-562429041bc2",
    "timestamp": "2020-02-03T21:50:47.743Z",
    "source": "agent",
    "version": "20150910",
    "webhookStatus":
    {
      "webhookEnabledForAgent": true
    },
    "agentEnvironmentId":
    {
      "agentId": "ecc89cbf-b744-4fa3-8c1f-bfef95fad4ad",
      "cloudProjectId": "leeme-vpnrki"
    }
  }
}

```

## 4.3 }Implementación

En este apartado se va a explicar cómo se ha realizado la aplicación en detalle, la forma en la que se ha tenido que tratar la creación del asistente virtual, así como las dificultades encontradas y las soluciones tomadas. Posteriormente se detallará la creación de la aplicación en Android studio junto con la base de datos que la complementa.

### 4.3.1. Implementación del asistente virtual

Para la implementación del asistente virtual no se requiere de mayor entorno que el propio que nos proporciona la herramienta en la nube. Se han creado dos Intents que se sumarán a los dos que se crean por defecto al crear un proyecto. Cuando se crea el proyecto se deben detallar las características del mismo, prestando especial atención al lenguaje en el que se quiere

realizar ya que algunos de ellos no están disponibles o se encuentran en fase beta. En el caso del español nos encontramos con una fase beta que, aunque cumple con las funcionalidades de envío de la información identificada, no siempre comprende adecuadamente las tildes en la pronunciación.

Una vez creado el proyecto es necesario asegurarse que la API que nos va a permitir usar es compatible con las herramientas de desarrollo elegida. Para realizar esta acción se han consultado los documentos de la propia herramienta, concluyendo que hoy en día sólo existe una versión de la API y sí es compatible con Android Studio y la versión 9.0. Uno de los problemas encontrados en el desarrollo del asistente consistía en la actualización del entorno de desarrollo que provocaba la pérdida de las funciones necesarias para las conexiones con el asistente; la solución realizada consistió en la descarga e instalación de la versión antigua de la herramienta de desarrollo.

Es muy importante tener en cuenta la prioridad que se le dan a los Intents en especial en el caso de este proyecto, ya que no se va a añadir contextos que permitan una mejor identificación del Intents en todos ellos, sino que es el propio Intents en sí. Esto provoca que se den algunos fallos de tildes que se entienden mal y por tanto se recibe una respuesta errónea, también provoca que este Intents en concreto no pueda favorecerse de la herramienta de aprendizaje, ya que podrían añadirse frases que no sean exactas y la lectura se realizase mal, obteniendo una respuesta favorable. Como solución para todos estos problemas se ha planteado la priorización de la que sí contiene contexto, pero no soluciona el problema ya que es generado en el propio Intents frase por la estructura que se requiere y la fase beta del lenguaje en cuanto a comprensión.

La aplicación requiere de un reconocimiento de la acción inicial que se quiere realizar por voz al que le sea aplicable contexto, por esa razón dentro de las frases de prueba se han introducido las que se han considerado que serían más frecuentes y se analiza sólo la acción como variable. De esta forma se pretende que el asistente virtual detecte la variable de las acciones leer, cargar y continuar dentro de diversas frases, haciendo que el asistente considere que las frases le pueden llegar con la variable en distintas posiciones, por sí sola o en frases que contengan esas palabras. Cabe destacar que, aunque en los casos que se han introducido, no se han producido errores, esto puede ocasionar que una frase, perteneciente a algún cuento, que contenga alguna de estas palabras, puede ser detectada como acción inicial. Estos errores se han tratado de solventar mediante la herramienta de aprendizaje que nos proporciona Dialogflow.

Cada Intent lleva asociada una respuesta o respuestas que se quieren proporcionar al usuario cuando se detectan. En estas respuestas podemos incorporar parte de la información que recogemos como se ha hecho en el caso de Accionini donde se ponen distintos verbos que nos dan a entender que se accede a la acción entendida, esta acción se guardará en una variable y posteriormente podrá ser usada. Las variables constituyen una parte fundamental de la creación de un asistente virtual, aunque en este caso sólo se han creado dos, permite analizar de forma lingüística las características que puede tener una misma palabra en distintos contextos. Por defecto se tiene una variable de la que extienden el resto, el resto de las variables hacen referencia a características de la frase como podría ser cantidad, tiempo, modo... que permiten hacer un análisis más profundo de las oraciones. Para este proyecto se han creado dos variables

extendidas de any, una de ellas se ha llamado frase y se han guardado todas las frases enteras de los cuentos como variables de este tipo y otra variable acción que sólo guarda leer, cargar y guardar, pero se sitúan en diferentes contextos y se extraen de frases más grandes como “quiero leer un libro”, donde se extraería que la información relevante y a guardar en la variable es leer.

Ninguno de los Intents expuestos en el párrafo anterior ha sido evaluado como final de conversación, queriendo mantener la conversación para que se pueda seguir leyendo. Cabe destacar que en algunos casos la herramienta cierra la conexión por sí sola y no se ha conseguido identificar el problema, se cree que es un bug.

Para realizar las conexiones del asistente virtual con la aplicación se requiere el token que nos proporciona el proyecto y realizar las clases necesarias en la aplicación Android, el resto será gestionado por la API y por la propia aplicación teniendo en cuenta los datos que va a proporcionar la misma. Al ser un servicio en la nube escala horizontalmente en función de las conexiones y con el mismo token se pueden conectar varios dispositivos sin problema por lo que esta información será estática para todo el proyecto.

#### **4.3.2. Implementación de la aplicación y base de datos**

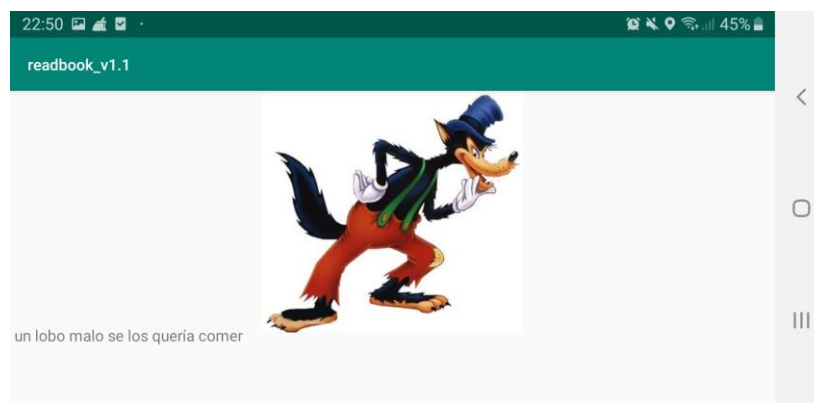
Se ha creado una clase MainActivity que contiene el código para cuatro botones. El primero de ellos que se muestra en pantalla nos permite leer, lo que nos llevará a una instancia de la clase Lectura cuando sea pulsado. En el segundo botón encontraremos la opción de continuar, donde podremos retomar el cuento en el punto en el que dijimos guardar por última vez. Para la creación de la acción de este botón se ha reutilizado la de leer, ya que la clase Lectura almacena en el archivo *preferencias* los valores cuento y contador cuando se dice guardar, por lo que en el momento en el que leamos volveremos al último punto de guardado. El tercer botón trata de darnos a elegir entre múltiples cuentos, como mejora de la aplicación. El cuarto botón establece la conexión con el asistente virtual, evaluando si la respuesta que se recibe es una Accionini o no, en caso de serlo ejecuta la acción y en caso de no serlo notificará que no ha entendido lo que se desea.



**Fig. 14. Interfaz MainActivity**

La clase lectura también establece al comienzo, en el `onCreate` las conexiones necesarias para vincularse al asistente virtual. En el caso de esta clase no se tiene que activar manualmente el control por voz, sino que se activa automáticamente mediante un botón invisible que se irá pulsando tras acabar una acción. Cabe destacar que en este caso la respuesta que se da debe ser corta, ya que si no puede perjudicar al reconocimiento de la siguiente frase. Cuando la frase que detecta el asistente virtual coincide con la que nos devuelve la base de datos y ambas con el cuento y página correcto, se accede a la base de datos y se incorpora la siguiente imagen y texto. Cuando en vez de leer la frase que se ve por pantalla el usuario decide que desea guardar y lo dice en voz alta, la clase guarda la página y cuento en el archivo *preferencias*. Esta clase cuenta con un manejador de la base de datos ya que contendrá toda la información necesaria respecto a los cuentos.





**Fig. 15. Interfaz Lectura**



**Fig. 16. Interfaz Lectura evolución**

La clase DBHelper nos permite crear la base de datos, contiene un método onCreate que crea las tablas si no existen y si es necesario elimina las que existían para poner las actualizadas. Esta clase cuenta con métodos que nos devolverán el nombre de la imagen asociada al cuento y página que se le esté pasando, otro método que realizará la misma acción y uno que creará un array de cuentos para dar la opción de realizar la mejora de leer múltiples cuentos. En el caso de esta clase se han encontrado problemas, ya que cada mínimo cambio en ella, lleva una subida del número de versión y al comenzar a trabajar por primera vez con la herramienta SQLite esto se desconocía.

Las clases Cuento y Contenido son clases equivalentes al almacenamiento de cada una de las tablas de la base de datos con el fin de perfeccionar el tratamiento en caso de ser necesario. Estas clases tienen los mismos atributos por lo que se puede hacer que la base de datos acceda una única vez para extraer los cuentos en caso de querer realizar la mejora de múltiples cuentos para leer y se puede disponer de un array de elementos que tratar, eliminando tiempos de espera en caso de que la aplicación creciese mucho en cuanto a contenido de la base de datos.

Cada una de las clases explicadas anteriormente y en las que se han mostrado interfaces gráficas tienen asociado un layout donde se realizan los cambios, de ser necesario para la visualización de la información. Uno de los mayores problemas encontrados ha sido en el layout de la clase Lectura, ya que requería del id de la imagen y no del nombre para la modificación de la interfaz, razón por la que se ha decidido guardar el nombre en la base de datos y las imágenes

en la carpeta drawable, ya que al ser guardadas en esta carpeta automáticamente se les asocia un id que puede recuperarse con el nombre de la imagen y el siguiente método:

```
getResources().getIdentifier(  
    biblioteca.imagenCuento(cuento, contador), "drawable",  
    getPackageName())
```

## 4.4 Planificación del proyecto

El proyecto se ha llevado a cabo haciendo un análisis de los requisitos que se consideraban necesarios y creando tareas que se han ido gestionando mediante metodología Kanban, agregando prioridades a las tareas, así como tiempo aproximado de realización:

- Creación de una herramienta que sea capaz de interpretar la voz y transcribirla:

- Estudio de las herramientas para desarrolladores del mercado
- Elección de las herramientas a utilizar
- Creación de la estructura de prioridades entre Intents
- Estructuración de Intents
- Creación del Intent Accionini
- Creación del Intent Frase
- Integración de Dialogflow con Android Studio

- Creación de la aplicación Android:

- Estudio de las compatibilidades con el resto de las herramientas
- Creación de un diagrama de flujo de datos
- Estructuración de la aplicación
- Creación de la clase MainActivity
- Creación de la interfaz de MainActivity
- Creación de las conexiones con Dialogflow
- Comprobación del traspaso de información con Dialogflow
- Creación de la acción del botón leer
- Creación de la clase Lectura
- Creación de las conexiones con Dialogflow
- Creación de la escucha activa de la clase Lectura
- Creación de la interfaz de la clase Lectura

- Creación de las conexiones con la base de datos
  - Comprobación de la correcta comunicación con la base de datos
  - Estudio de las múltiples opciones para guardar el progreso
  - Elección de opción de guardado de proceso
  - Implementación del guardado de proceso
  - Creación de la acción continuar aplicando proceso de guardado
- Creación de la base de datos:
    - Estudio de las herramientas de almacenamiento de datos para Android Studio
    - Elección de herramienta de almacenamiento
    - Decisión de datos a almacenar
    - Estructuración de los datos
    - Creación de la base de datos
    - Creación de la tabla Contenidos
    - Creación de la tabla Cuentos
    - Poblar la tabla Contenidos
    - Poblar la tabla Cuentos
    - Creación del método para devolver la frase
    - Creación del método para devolver la imagen
    - Creación del método para devolver los cuentos
    - Creación de las conexiones con la clase Lectura
- Gestión del estudio:
    - Citación con los expertos en educación infantil
    - Creación del documento concretando el uso de las propiedades intelectuales
    - Citación para el traspaso de propiedades intelectuales
    - Citación para obtención de resultados
- Creación de la documentación:
    - Búsqueda de las plantillas
    - Recopilación de documentación usada
    - Redacción de los distintos apartados
    - Creación del estudio

- Tareas de mejora:
  - Permitir continuar por donde se dejó de leer y se decidió guardar
  - Modificación de las frases para mayor comprensión del asistente virtual
  - Simplificación de la interfaz
  - Incorporación de más cuentos
  - Permitir elegir cuento que leer
  - Buscar forma de incorporar cuentos propios
  - Creación de un manual de usuario

Dada la disponibilidad que era de aproximadamente tres horas al día se ha contado como un punto de duración cada tres horas. Se le ha dado prioridad al desarrollo del asistente virtual, posteriormente a la creación de la aplicación y menor prioridad al almacenamiento y documentación. Las tareas de mejoras han sido algunas realizadas y otras estudiadas sin poder llevarlas a cabo. Estas tareas se encontraban fuera del planning para la realización del proyecto ya que se consideran posibles mejoras y se querían llevar a cabo en caso de contar con el tiempo necesario para desarrollarlas.

Los principales requisitos que se han evaluado a la hora de realizar el proyecto son los siguientes:

- Interfaz simple
- Interfaz llamativa
- Fácil control de la aplicación
- Frases simples
- Frases cortas
- Funcionalidad de corrección de lectura
  - Funcionalidad de manejo mediante voz

## 5. Estudio de la herramienta en colegios

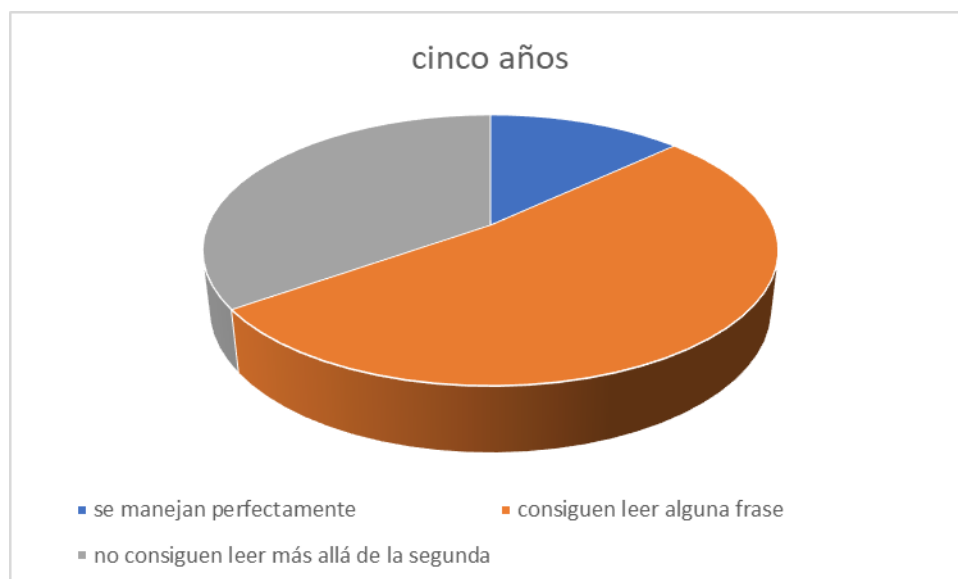
Una de las características que hace especial este proyecto es que ha sido autorizado al uso de cuentos que hoy en día se están usando en el colegio Calasanz de Alcalá de Henares para enseñar a leer a los niños. Estos cuentos han sido realizados por maestras de esta institución, con conocimientos suficientes como para aconsejar respecto a su campo de estudio.

Durante todo el desarrollo de la aplicación se ha ido consultando y se han ido modificando las cosas para que sean lo más útiles a futuro posible.

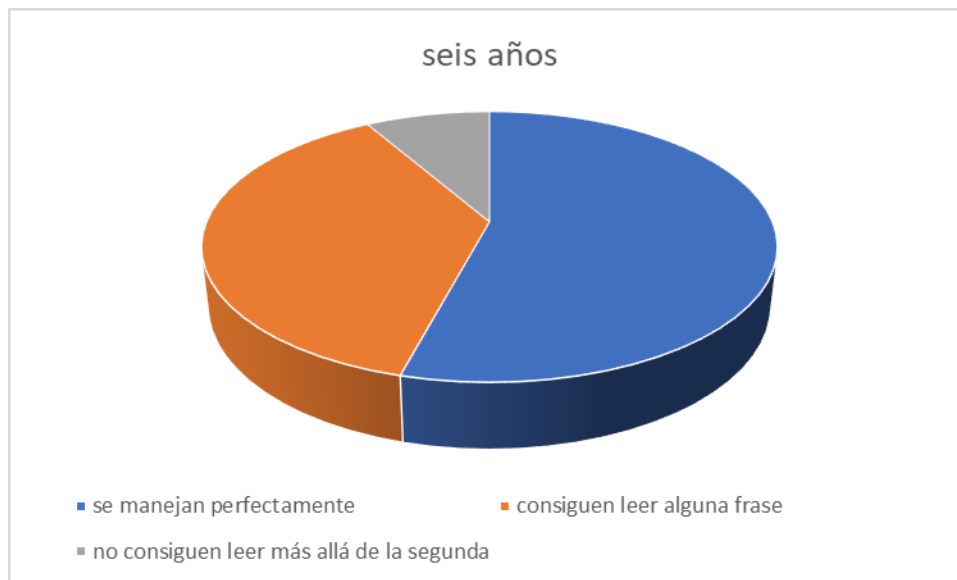
Debido a la falta de tiempo para la realización de este estudio, no se han podido realizar todas las pruebas que se deseaban en un origen. Entre algunas de ellas estaban la posibilidad de medir la mejora de la lectura tras el uso de la aplicación durante periodos de tres meses, ya que se requiere un periodo de familiarización con la herramienta, así como de margen para ver la mejora o la creación del hábito de estudio. También se ha tenido que eliminar el análisis del porcentaje de flujo de datos que se destina a leer mediante las herramientas de análisis de Dialogflow, analizando las duraciones de las conversaciones y con ello el tiempo de uso de la aplicación en general.

La decisión que se tomó fue la prueba en diversos niños para ver si podían utilizar la aplicación, los tiempos eran los adecuados para las edades de los niños que las usaban y así calcular de forma estadística a partir de qué años se podría usar la aplicación.

La muestra de datos se toma durante el primer trimestre del año escolar, a finales de este. Se evalúan entre alumnos de cinco años que acaban de empezar a leer frases enteras, alumnos de seis años que ya tienen algo más de fluidez en la lectura y alumnos de siete años que ya leen.



**Fig. 17. Representación datos niños de cinco años**



**Fig. 18. Representación datos niños de seis años**



**Fig. 19. Representación datos de niños de siete años**

Con estos resultados podemos concluir que el asistente virtual no reconoce fácilmente las voces infantiles y además la aplicación a pesar de estar diseñada para niños de cinco años, acorde con el aprendizaje de la lectura, le falta ser perfeccionada para que sea destinada al público que se quiere. Es muy probable que, en gran parte, la dificultad a la hora de leer no sólo se deba a la interpretación del asistente virtual, sino que también se deba al tiempo que se deja para la lectura de la frase. Tras la recogida de los datos se ha intentado incrementar el tiempo que transcurre desde que el micrófono comienza a grabar hasta que termina y envía la frase al asistente virtual, pero no se ha conseguido.

Estos datos nos revelan que la cantidad de mejoras que se pueden aplicar a futuro, tras la mejora de las herramientas de reconocimiento de voz y la interconexión entre las herramientas de desarrollo, realmente harían que la aplicación se acercase más a su objetivo inicial que era ser un complemento para el aprendizaje de la lectura y una herramienta de ayuda para la creación de un hábito de lectura.

Cabe la posibilidad de que si se realizase el mismo estudio tras dejar un periodo mayor de adaptación de los niños a la aplicación, así como familiarización del uso de la misma, los datos cambiasen significativamente. Debido al tiempo para el desarrollo del estudio, sólo se ha podido recoger una única tanda de datos estadísticos muy simples que nos permiten hacernos a la idea de qué puede estar fallando en el diseño y cómo mejorarlo.

## 6. Conclusiones

La conclusión más evidente de este proyecto es que las herramientas que se necesitaban para la creación de los asistentes virtuales y los desempeños que se necesitaban de las mismas no se han podido cubrir en su totalidad. El asistente no diferencia bien entre palabras con tildes y sin ellas como es “tomo” y “tomó”, este problema ha generado tanto pérdida de tiempo debido a la necesidad de modificación de las frases de los cuentos por frases mucho más sencillas como pérdida de valor de la propia aplicación, ya que las tildes y la diferenciación entre ellas es algo necesario a la hora de aprender a leer en castellano.

Por otro lado, la sincronización entre herramientas también depende mucho de las plataformas y lenguajes lo que condiciona al desarrollador a elegir las plataformas y lenguajes compatibles. También contienen herramientas útiles como es el caso de los servicios que proporcionan medidas de datos de funcionamiento, así como de aprendizaje.

Otra conclusión a mi entender relevante es que un niño no puede leer cualquier cuento con cualquier frase. La posibilidad que se me ha brindado tanto de emplear herramientas de expertos como de recibir consejo de los mismos me ha hecho percatarme de la importancia de los colores, la simplicidad de las frases y la facilidad de manejo que deben tener este tipo de aplicaciones.

Como conclusión genérica se podría decir que la implementación de esta aplicación dentro de un par de años, cuando las herramientas que están en fase beta, cumplan con su labor y las herramientas disponibles permitan una mayor explotación de los asistentes virtuales, podría ser muchísimo mejor realizada y se le podrían agregar funcionalidades que a día de hoy se han planteado para el proyecto, pero ha sido imposible realizarlas.



## **7. Trabajo futuro**

Una de las ideas que se tenía cuando se comenzó este proyecto era poder añadir tú tus propios cuentos, cargándolos en la propia aplicación. Esta idea no ha sido posible realizarla a día de hoy porque el asistente requiere de la incorporación manual de las frases por lo que no se pueden incorporar de otra forma, pero quizá en futuras versiones sí se permita una carga automática de información.

Uno de los mayores problemas ha sido el aprendizaje del asistente virtual, ya que según la documentación aprende de la voz del usuario, pero a la hora de diferenciar tildes no es capaz. La creación de un asistente que sí que pueda interpretar correctamente el lenguaje y tono, ya fuese con redes neuronales o por la mejora de las herramientas existentes, haría de este proyecto un proyecto usable y además sería más probable que cumpliese con el objetivo inicial de la creación de este proyecto, fomentar el hábito de lectura y ayudar en el aprendizaje de la lectura para los niños.

Otra mejora sería hacer las imágenes con movimiento, esto llamaría la atención del niño más que con simples colores, lo que permitiría una mayor atención del pequeño y por tanto sería más útil para cumplir con el objetivo.

La creación de distintos niveles o de usuarios que guarden la complejidad a la que quieren leer haría de esta herramienta una posible ayuda para el aprendizaje de idiomas en caso de poder añadir múltiples idiomas, ya no sólo se restringiría el público a un entorno más bien infantil, sino que abriría el mercado a toda clase de edades.

## 8. Coste del proyecto

Para la elaboración de este proyecto se ha necesitado un ordenador con las herramientas Android studio, SQLite y Dialogflow, para poder ejecutar la última se requiere de una conexión a internet. También se ha usado para la creación de la documentación, así como la búsqueda de información.

Se requiere de una persona para el desarrollo de la aplicación, del asistente virtual y la interconexión entre ambas herramientas.

En cuanto a las infraestructuras necesarias se requiere al menos una conexión a la red eléctrica que se tendrá en cuenta a la hora de evaluar el presupuesto.

El proyecto se ha planteado para una duración de once meses, teniendo un retraso de un mes debido a la indisposición del único desarrollador que trabaja en dicho proyecto durante el transcurso de este.

	1 mes	total	con empleado real	en caso de venta
luz	20,00 €	240,00 €	240,00 €	240,00 €
ordenador	800,00 €	800,00 €	800,00 €	800,00 €
salario	- €	- €	9600	9600
coste herramientas de desarrollo	- €	- €	- €	- €
licencias de software				500
nº ventas al mes				5
ganancias por licencias	- €			30000
ingresos	- €	- €	- €	30.000,00 €
gastos	820,00 €	1.040,00 €	10.640,00 €	10.640,00 €
beneficios	- 820,00 €	- 1.040,00 €	- 10.640,00 €	19.360,00 €

**Tabla 1. Representación de costes**

Estamos suponiendo que en el caso de la luz se paga sólo lo que se gasta aproximadamente, en el caso del salario son ochocientos euros al mes y que el ordenador tiene el mismo precio que el ordenador que realmente se ha estado usando para el desarrollo del proyecto. También se han añadido dos casos hipotéticos en los que se añade un empleado con sueldo real y en el que se añaden unas posibles ventas.

Cabe destacar que el proyecto en sí no tiene costes de herramientas de desarrollo porque las herramientas y las funcionalidades que se han empleado en la creación del proyecto son de uso gratuito por lo que no se puede añadir coste alguno en este campo.

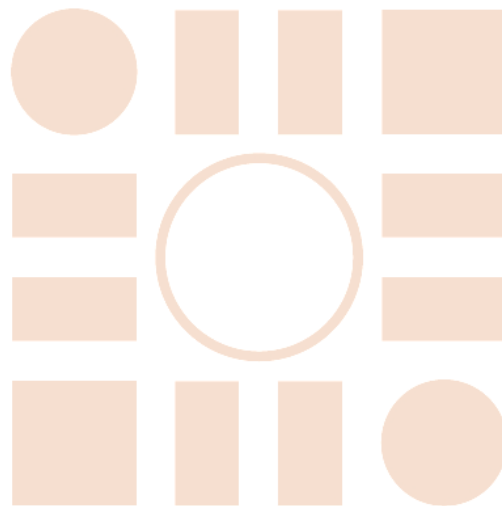
El cálculo de los costes se ha hecho para los doce meses que ha durado el proyecto, lo que quiere decir que se ha incorporado el coste del retraso en el presupuesto total.

## 9. Bibliografía

- Aplicaciones ya existentes en el ámbito : <https://www.julianmarquina.es/10-aplicaciones-para-aprender-a-leer-y-practicar-la-lectura-a-traves-del-smartphone-o-tablet/>
- Acceso a documentación de Android para el desarrollo: <https://developer.android.com/reference/java/>
- Consulta de errores concretos y soluciones: <https://stackoverflow.com/>
- Documentación para la creación de la base de datos: [https://www.tutorialspoint.com/android/android\\_sqlite\\_database.htm](https://www.tutorialspoint.com/android/android_sqlite_database.htm)
- Creación del asistente virtual: <https://dialogflow.com/docs/integrations/actions>
- Funcionamiento y estructura interna del asistente virtual a utilizar: <https://developers.google.com/assistant/app/overview>
- Información de los asistentes virtuales descartados para el desarrollo: <https://bespoken.io/blog/crear-skill-de-alexa/>
- Herramientas de aprendizaje con asistentes virtuales <https://www.educaciontrespuntocero.com/tecnologia/asistentes-inteligentes-en-educacion/>
- Herramientas con el asistente de Google para el aprendizaje: <https://assistant.google.com/explore/c/3/?hl=es-ES>



Universidad de Alcalá  
Escuela Politécnica Superior



ESCUELA POLITECNICA  
SUPERIOR



Universidad  
de Alcalá